IN THE TITLE

Please change the title to the following:

OVERRIDING DEPENDENCY CHECKING LOGIC AND HANDLING EXCEPTIONS

IN THE ABSTRACT

Please replace the abstract with the abstract attached herein on a separate page.

IN THE CLAIMS:

Claims 1-20 (Canceled Herein)

21. (Newly Added) A system comprising:

main memory storing an instruction set; and

a processor operably connected to main memory by a bus network, wherein the processor comprises:

a floating point unit;

a register stack;

dependency checking logic for determining whether instructions are executed sequentially or in parallel;

two execution units for executing instructions; and

ROM storing a micro-code handler that is invoked when two move instructions operating on registers in the register stack are executed in parallel and cause a stack underflow exception, the micro-code handler being operable to overcome any dependency between the registers and allow execution of the move instructions in parallel without a stack underflow exception.

22. (Newly Added) The system of claim 21, wherein the micro-code handler is operable to flush the two move instructions if any of the two move instructions causes the stack underflow exception.

23. (Newly Added) The system of claim 21, wherein the micro-code handler is operable to replace contents of one of the registers with a QNaN in response to detecting the stack underflow exception.

24. (Newly Added) The system of claim 21, wherein the execution units are operable to execute the move instructions in parallel such that first data of a first register of the registers is moved to a second register of the registers, and second data of the second register is substantially simultaneously moved to the first register.

25. (Newly Added) The system of claim 21, wherein the first register is a top register in the floating point stack and the second register is another register in the floating point stack.

26. (Newly Added) The system of claim 24, wherein the register stack is for the floating point unit.

27. (Newly Added) In a processor based computer system having dependency checking logic and a register stack, a method comprising:

overriding the dependency logic such that move instructions associated with the stack registers are operable to be executed in parallel;

executing the move instructions in parallel;

determining whether a stack underflow exception has occurred and if it has;

flushing the move instructions; and

invoking a micro-code handler algorithm that operates to allow execution of the move instructions in parallel without a stack underflow exception.

28. (Newly Added) The method of claim 27, wherein the register stack comprises a top register and a selectable register, and wherein invoking the microcode handler further comprises:

determining whether the top register of the stack is empty; and

replacing contents of the top register with a defined architectural response in response to said top register being empty.

29. (Newly Added) The method of claim 28, wherein determining whether the top register of the stack is empty, further comprises:

replacing the contents of the selectable register with the defined architectural response in response to the top register of the stack being not empty.

30. (Newly Added) The method of claim 29, further comprising:

executing the move instructions in parallel if the top or selectable register contents have to be replaced with the defined architectural response.

31. (Newly Added) The method of claim 28, further comprising:

replacing the contents of the selectable register with the defined architectural response if the contents of the top register have been replaced with the defined architectural response and a stack underflow exception has occurred.

32. (Newly Added) A processor, comprising:

dependency checking logic; and

a register stack having a top register and a plurality of other registers; wherein the processor is configured to:

override the dependency logic such that operations related to the register stack are operable to be executed in parallel;

substantially simultaneously switch contents of the top register of the register stack with contents of another register of the register stack and vice versa, such that an error does not occur; and

execute an algorithm that replaces the contents of the top register, the other register, or both the top and other registers with a defined architectural response if an exception occurs when the content of both registers are switched.

33. (Newly Added) The processor of claim 32, wherein the defined architectural response is a QNaN.

34. (Newly Added) The processor of claim 32, further comprising:

at least two execution units operable to substantially simultaneously switch the contents of the top register of the register stack with the contents of the another register of the register stack and vice versa.

35. (Newly Added) The processor of claim 34, wherein the at least two execution units are provided in at least one pipelined circuit.

36. (Newly Added) The processor of claim 34, further comprising a floating point unit.

37. (Newly Added) The processor of claim 36, wherein the register stack stores data for the floating point unit.

38. (Newly Added) The processor of claim 32, further comprising a ROM storing a micro-code that includes the algorithm that replaces the contents of the top register, the other register, or both registers with a defined architectural response if an exception occurs when the content of both registers are switched.

39. (Newly Added) An apparatus comprising:

at least two execution unit means for executing instructions;

dependency checking means for checking the dependency of data associated with instructions executed by the at least two execution unit means;

register stack means for storing data and including a top register and a plurality of other registers;

switching means for substantially simultaneously switching contents of the top register of the register stack means with contents of another register of the register stack means and vice versa, such that an error does not occur; and

execution means for executing an algorithm that overrides the dependency checking means such that instructions related to the register stack means are operable to be executed in parallel and that replaces the contents of the top register, the other register, or both the top and other registers with a defined architectural response if an exception occurs when the content of both registers are switched.

40. (Newly Added) The apparatus of claim 39, further comprising a floating point unit means for executing floating point operations and the register stack means stores data for the floating point operations.